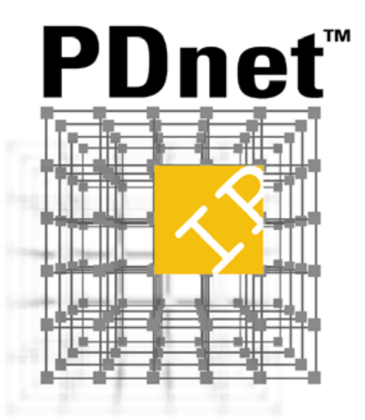


PAD-740e Benutzerhandbuch



automation technologies GmbH

Einschränkung der Gewährleistung:

Die Angaben und Hinweise in diesem Handbuch sind sorgfältig geprüft und entsprechen den verfügbaren Daten zur Zeit der Drucklegung. Es wird keine Garantie für die Richtigkeit des Inhaltes übernommen.

Die im Handbuch verwendeten Software- und Hardware-bezeichnungen sind zum Teil eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Herausgeber:

APEX automation technologies GmbH
Vossenkamp 4
38104 Braunschweig

Telefon 0531-3704-0
Telefax 0531-3704-299

http://www.apex.de



Der PDnetIP-Controller erfüllt die Anforderungen der EU-Richtlinie 89/336/EWG „Elektromagnetische Verträglichkeit“.
Die Konformität des PDnetIP-Controllers mit der o.g. Richtlinie wird durch das CE-Zeichen bestätigt.

INHALT

0 INHALT
1.1 Produktübersicht

1 Einleitung

Dieses Handbuch unterstützt Sie beim Einbau, der Konfiguration und der Inbetriebnahme des PDnetIP-Controllers PAD-740e. Es beschreibt die Installation folgender Produktvarianten:

- **PAD-740e** verfügbar als
  PAD-740e/T, PAD-740e/L

Die APEX automation technologies GmbH behält sich Änderungen und Weiterentwicklungen vor, die zu Abweichungen von den in diesem Handbuch angegebenen Daten führen können.

Die Hardware wird von der APEX automation technologies GmbH zusammen mit einer PDnetIP-CD geliefert welche die für den Betrieb des PDnetIP-Controllers notwendige Software und Dokumentation beinhaltet.

Bestandteil der PDnetIP-CD ab Version 2.41 ist das Dokument „PDnetIP Versionsübersicht“. Das Dokument gibt einen Überblick über die einzelnen Versionen.

Benutzen Sie die Software und Dokumentation einer PDnetIP-CD’s immer als Einheit.

Die aktuelle PDnetIP-CD kann unter der Adresse ftp2.apex.de abgerufen werden. Benutzen Sie für den Download den Benutzernamen `kunde` und das Passwort `apex`.

1.1 Produktübersicht

Der PDnetIP-Controller PAD-740e dient zum Anschluss einer Simatic S7-400 an das PDnetIP. Der Datenaustausch zwischen Anwenderprogramm und PDnetIP-ControllerP wird über den Funktionsbaustein FB62 und die Funktionen FC211, FC213, FC216 abgewickelt.

Um den unterschiedlichsten Anforderungen gerecht zu werden, stehen verschiedene Versionen des PAD-740e zur Verfügung. Die einzelnen Versionen unterscheiden sich hinsichtlich der Ausführung der PDnetIP-Anschlüsse.

Übersicht der PAD-740e

Artikel Nummer	Name	Beschreibung
10102020	PAD-740e/T	PDnetIP - Controller für Siemens S7-400, mit RJ 45-Buchse
10102022	PAD-740e/L	PDnetIP - Controller für Siemens S7-400, mit RJ 45-Buchse und MTRJ-Anschluss

Der PDnetIP-Controller ist in verschiedenen Versionen erhältlich. Ergänzend zu den in der oberen Tabelle dargestellten Versionen, gibt es weitere optional erhältliche Module. Die folgende Tabelle gibt eine Übersicht, welche Protokolle lizenzrechtlich wann zur Verfügung stehen. Ob die gewünschte Kombination von Protokollen gleichzeitig nutzbar sind, ist separat zu prüfen.

<table>
<tr><th>Protokolle</th><th>Anmerkung</th></tr>
<tr><td>ModBus/TCP</td><td rowspan="2">Steht in jedem PDnetIP-Controller zur Verfügung.</td></tr>
<tr><td>VDM (PDnetIP)</td></tr>
<tr><td>RFC1006</td><td>Separat zu erwerbende Lizenz</td></tr>
<tr><td>TCP-Schnittstelle</td><td>Separat zu erwerbende Lizenz</td></tr>
</table>

Der PDnetIP-Controller ist mit Leitungsinterfacemodulen (LIM) bestückt, welche die Leitungsart der seriellen Schnittstellen anpassen.

Die Lieferung erfolgt mit bestücktem LIM-232. Die Module können nicht durch den Anwender auf dem PDnetIP-Controller gewechselt werden. Soll der PDnetIP-Controller mit einer anderen Leistungsphysik arbeiten, so ist dies bereits bei der Bestellung zu berücksichtigen.

Übersicht der Leitungsinterfacemodule

Artikel Nummer	Name	Beschreibung
10213010	LIM-232	LIM-232 Leitungs-Interface-Modul für RS-232 ohne Potentialtrennung
10213011	LIM-422	Leitungs-Interface-Modul für RS-422 ohne Potentialtrennung
10213012	LIM-485	Leitungs-Interface-Modul für RS-485 ohne Potentialtrennung
10213016	LIM-20 mA	Leitungs-Interface-Modul für 20 mA Stromschleife mit Potentialtrennung

1.2 Bestimmungsgemäße Verwendung

Der PDnetIP-Controller PAD-740e ist ausschließlich dazu bestimmt:

- eine SIMATIC S7-400 mit dem PDnetIP zu verbinden.

Wird der PAD-740e anders als in diesem Handbuch beschrieben verwendet, ist kein sicherer Betrieb gewährleistet.

Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller, sondern der Betreiber verantwortlich.

1.3 Allgemeine Hinweise

Der Einbau des PDnetIP-Controllers sollte von einer Fachperson durchgeführt werden. Voraussetzung für einen sicheren Einbau ist ein antistatischer Arbeitsplatz. Die APEX automation technologies GmbH trägt keine Verantwortung für Schäden, die durch einen unsachgemäßen Einbau entstehen und/oder entstanden sind.

Das vorliegende Benutzerhandbuch erläutert den Einbau und die Inbetriebnahme der PDnetIP-Controller. Informationen über den Umgang mit dem Computersystem bzw. dem (den) Betriebssystem(en), die im Zusammenhang mit diesem Computersystem betrieben werden sollen, sind den entsprechenden System-Handbüchern zu entnehmen.

Nachdem die Verpackung geöffnet und dieses Handbuch, sowie der PDnetIP-Controller herausgenommen wurden, sollte der PDnetIP-Controller bis zu seinem Einbau in dem Antistatik-Schutzbeutel verbleiben.

1.4 Lieferumfang

Zum Lieferumfang der PAD-740e gehören:

1. PAD-740e in Antistatikverpackung.
2. CD-ROM PDnet + PDnetIP.
3. Dieses Benutzerhandbuch

2 Technische Daten

Zuordnung	
Zentralbaugruppen	CPU 412, CPU 414, CPU 416, CPU 417
Anschluss	Einfach breiter Steckplatz im Zentral-Rack
Versorgung	
Betriebsspannung	5 V / 900 mA aus dem Zentral-Rack
PDnetIP-Anschlüsse	
Anzahl	1
Übertragungs-geschwindigkeit	10/100 Mbit/Sekunde Ethernet 10Base-T, 100Base-TX, 100Base-FX (LWL)
Kabeltyp	Twisted-Pair mit RJ-45 Stecker Cat. 5 LWL mit MT-RJ Anschluss
Serielle Schnittstellen	
Anzahl	2
Leitungsphysik	Wählbar über Steckmodul: RS-232, RS-485 oder RS-422
Übertragungs-geschwindigkeit	50 — 115.200 Bit/Sekunde
Anschluss	RJ-45 Buche
Prozessor	
Typ	Intel 386EX
Speicher	2 MB CMOS-RAM und 1 oder 2 MB Flash-EPROM
Mechanischer Aufbau	
Kartenformat	S7-400 Modulbecher
Breite	einfache Breite
Gewicht	ca. 450 g
Umgebungsbedingungen	
Temperatur	0 — 60 °C (im Betrieb), -40 — +85 °C (Lagerung)
Luftfeuchtigkeit	10 % — 80 % (nicht kondensierend)

2.1 Hardware

2.1.1 Ethernet

Der PAD-740e/L ist mit zwei Schnittstellen ausgestattet. Er besitzt sowohl einen RJ45-Anschluss als auch ein MT-RJ-Anschluss. Zu einem Zeitpunkt kann nur ein Anschluss genutzt werden. Wird der PDnetIP-Controller über den MT-RJ-Anschluss betrieben, so muss auf der Gegenstelle als Betriebsart Halb-Duplex konfiguriert werden.

Die Mac-Adresse (Ethernet-Adresse) des PDnetIP-Controllers ist in der Hardware gespeichert und unveränderlich. Alle gelieferten PDnetIP-Controller nutzen eine mit **`00:05:DA`** beginnende Mac-Adresse.

Wichtig: Es wird empfohlen, den PDnetIP-Controller im Halbduplex-Mode zu betreiben. Im Vollduplex-Mode gibt es Kollisionszustände, welche nicht erkannt werden können, was zu Telegrammwiederholungen führt. Die Ursache liegt darin, dass auch in einem geswitchten Ethernet Kollisionen entstehen können.

Die folgende Tabelle zeigt die Belegung der RJ-45 Anschlussbuchse, die der IEEE802.3 Twisted Pair-Schnittstelle entspricht.

Pin	Signal
1	Transmit +
2	Transmit -
3	Receive +
4	-
5	-
6	Receive -
7	-
8	-

2.1.2 Serielle Schnittstellen

Der PAD-740e ist mit 2 seriellen Schnittstellen ausgestattet. Die Schnittstellen sind als 8-polige RJ-45 Buchsen auf der Frontseite ausgeführt.

Die Leitungsinterfacemodule (LIM-Module) passen die seriellen Schnittstellen des PDnetIP-Controllers an die Leitungsphysik an. Die Standardausstattung erfolgt mit bestücktem LIM-232.

Die LIM-Module können nicht durch den Anwender auf dem PDnetIP-Controller gewechselt werden.

Die folgende Tabelle zeigt die PIN-Belegung der seriellen Schnittstellen in Abhängigkeit der eingesetzten LIM-Module.

PIN Belegung der seriellen Schnittstellen

Pin	LIM-232 V.24 Signal	LIM 485 RS 485	LIM-422 RS 422
1	CTS	TxD -	TxD -
2	DSR	—	—
3	RxD	TxD +	TxD +
4	GND	—	—
5	GND	GND	GND
6	TxD	RxD +	RxD +
7	DTR	+5V	+5V
8	RTS	RxD -	RxD -

— nicht belegt

Die folgende Tabelle zeigt, wie ein Programmierkabel zu konfektionieren ist, um den PDnetIP-Controller über die serielle Schnittstelle mit NetPro oder DLSE32 laden zu können.

PDnetIP-Controller	Programmiergerät (PC)
6 (TxD)	2 (TxD)
3 (RxD)	3 (RxD)
5 (GND)	5 (GND)

2.2 Software

Der Leistungsumfang der PDnetIP-Controller wird im wesentlichen von der verwendeten Firmware bestimmt. Wird die Firmware auf dem PDnetIP-Controller getauscht, ist das entsprechend für diese Firmware gültige Handbuch zu nutzen.

Um den RFC1006 Dienst oder die TCP-Schnittstelle nutzen zu können, müssen diese auf der Karte freigeschaltet sein. Der nachfolgende Text geht immer von der Verfügbarkeit der entsprechenden Dienste aus.

2.2.1 Leistungsmerkmale

Protokoll	Limit – Maximum
ARP	64 ARP-Einträge
ICOS/TCP	32 Verbindungen
ModBus/TCP	32 Verbindungen
RFC1006	50 Verbindungen
TCP-Schnittstelle	50 Verbindungen

Folgende Limits gelten für alle Protokolle in Summe

- 128 TCP-Verbindungen
- 5 UDP-Verbindungen

Eine Reserve von 10% sollte berücksichtigt werden.

2.2.2 TCP/IP

Der PDnetIP-Controller nutzt die in der Tabelle aufgeführten Port-Nummern. Diese Informationen sind für den Netzwerk-Administrator von Bedeutung, wenn aktive Netzwerkkomponenten wie Router oder Firewalls genutzt werden.

Protokoll	Port-Nummer
DNS	TCP 53
BOOTP/DHCP	UDP 68
ISO TP0 over TCP (RFC1006)	TCP 102
ModBus/TCP	TCP 502

PdnetIP (VDM)	UDP/TCP 2843
ICOS/TCP	UDP/TCP 9876
TCP	Abhängig von der Projektierung

2.2.2.1 ModBus/TCP

Der PDnetIP-Controller unterstützt ModBus/TCP Funktionen im Server- und Master-Betrieb. Dazu benötigt der PDnetIP-Controller mit Hilfe des FB62 Zugriff auf den Signalspeicher der Steuerung.

Der PDnetIP-Controller kann zu einem Zeitpunkt nur eine limitierte Anzahl von ModBus/TCP-Verbindungen aufrecht halten. Eine Station (IP-Adresse) kann gleichzeitig mehrere Verbindungen zu den PDnetIP-Controllern unterhalten und dadurch eine bessere Kommunikationsgeschwindigkeit erreichen.

Nachfolgende Tabelle gibt einen Überblick der unterstützten ModBus/TCP-Funktionen:

- 01 (0x01) Read Coils
- 03 (0x03) Read Holding Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Register
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Registers

Über ModBus/TCP können folgende Datentypen der S7 adressiert werden.

- Merker
- Merkerbyte
- Merkerworte
- Worte
- Doppelworte
- Gleitkommaworte

Strukturen können mit ModBus/TCP nicht adressiert werden.

Für jeden Datenbereich muss zuvor mit NetPro ein Mapping definiert werden.

Ein Mapping ordnet den ModBus/TCP Datentypen Register bzw. Coils einen Speicherbereich der SPS zu. Dabei wird festgelegt, wie die Datentypen auf einander abgebildet werden und wie viele Elemente dieser Bereich umfasst. Zusätzlich kann jeder Bereich mit Eigenschaften wie "Read Only" oder Informationen zur Byte-Order (Word-Swap) konfiguriert werden. Für einen ModBus/TCP Dienst können mehrere verschiedene dieser Bereiche angelegt werden. Eine ModBus/TCP Transaktion, die sich auf Elemente bezieht, die nicht vollständig einem Mapping zugeordnet werden können, wird zurückgewiesen.

2.2.2.2 RFC1006 (ISO TP0 overTCP)

Der PDnetIP-Controller unterstützt optional RFC1006 bzw. „ISO TP0 over TCP“. Dazu benötigt er mit Hilfe des PDnet-Bausteins Zugriff auf den Signalspeicher der Steuerung.

Auf Netzebene werden die Daten gemäß ISO 8073 (TP0) über ein TCP/IP Netzwerk in Blöcken übertragen. Der PDnetIP-Controller überträgt die Netto-Daten direkt in den Signalspeicher der Steuerung.

Die RFC1006-Projektierungsinformationen werden als eine Liste von Verbindungen bearbeitet. Jede Verbindung besitzt folgende Eigenschaften:

- Eine Kennung (1..65535) sowie einen symbolischen Namen (24 Zeichen) zur eindeutigen Identifikation der Verbindung.
- Die IP-Adresse der Partnerstation, den eigenen TSAP sowie den TSAP der Partnerstation. Der TSAP kann eine Länge von 2 bis 16 Zeichen besitzen.
- Die genutzte Auftragsart (SEND, RECEIVE oder SEND und RECEIVE) sowie optionale Konfigurationsparameter. Eine Verbindung mit der Auftragsart SEND kann nur Daten

senden und keine Daten empfangen. Entsprechend kann eine Verbindung mit der Auftragsart RECEIVE nur Daten empfangen und keine Daten senden. Die Betriebsart SEND/RECEIVE ermöglicht eine bidirektionale Kommunikation.

- Die Adresslage des Datenpuffers im Signalspeicher der Steuerung. Die Adresslage wird dabei als absolute Signalspeicheradresse definiert.

Jede projektierte Verbindung belegt im Signalspeicher der Steuerung vier Merker, sowie vier Worte. Über diese Daten kommuniziert das Anwenderprogramm der SPS mit dem RFC1006-Treiber der Firmware.

Zur Ablage dieser Informationen ist vorab eine Startadresse im Wort- sowie eine Startadresse im Merkerbereich der Steuerung zu definieren. Die Adressen werden vom RFC1006-Treiber automatisch belegt.

Die folgenden Tabellen beschreiben die Bedeutung der für jede Verbindung belegten Status-Worte und Merker. Der Suffix ergänzt den mit NetPro projektierten Verbindungsnamen.

Suffix	Vorhanden	Zustand
'_ID'	Immer	Die mit NetPro projektierte Verbindungskennung.
'_STATE'		Der aktuelle Verbindungsstatus.
'_SB'	Sende-richtung	Beauftragungsbit in Senderichtung. Über das Sendebit zeigt das Anwenderprogramm der SPS an, dass ein Sendetelegramm vorliegt.
'_SQ'		
'_AWS'		Letzter/Aktueller Status in Senderichtung.
'_EB'	Empfangs-richtung	Das Empfangsbit informiert das Anwenderprogramm der SPS über ein neues Empfangstelegramm.
'_EQ'		
'_AWE'		Letzter/Aktueller Status in Empfangsrichtung.

Mögliche Zustände SB/SQ

SB	SQ	Zustand
0	1	Sendepuffer frei. Dieser Zustand wird vom PDnetIP-Controller im Einschaltmoment gesetzt, so dass im Anwenderprogramm der SPS keine Normierung der SB/SQ-Merker durchgeführt werden muss.
1	0	Wird vom Anwenderprogramm der SPS gesetzt, wenn ein Telegramm gesendet werden soll. Wurde der Sendevorgang angestoßen, darf dieser vom Anwenderprogramm der SPS nicht wieder gelöscht werden. Das Anwendungsprogramm der SPS hat die Kontrolle an den PDnetIP-Controller abgegeben und muss passiv auf die Reaktion warten. **Veränderung von SB/SQ ist nicht zulässig.**
0	0	Zustand ist nicht zugelassen und wird vom PDnetIP-Controller auf SB=0 und SQ=1 normiert.
1	1	

Ablauf Sendevorgang

Schritt	SB	SQ	Zustand
1	0	1	Der Sendepuffer ist verfügbar.
2	1	0	Das Anwenderprogramm der SPS hat den Sendepuffer mit Daten befüllt und zeigt über das gesetzte Sendebit den Sendewunsch an.
3	0	1	Der PDnetIP-Controller hat den Sendewunsch bearbeitet und der Sendepuffer steht wieder zur Verfügung.

Mögliche Zustände EB/EQ

EB	EQ	Zustand
0	0	Der Empfangspuffer ist frei. Dieser Zustand wird vom PDnetIP-Controller im Einschaltmoment gesetzt, so dass im Anwenderprogramm der SPS keine Normierung der EB/EQ-Merker durchgeführt werden muss.
0	1	Zustand ist nicht zugelassen und wird vom PDnetIP-Controller normiert.
1	0	Der PDnetIP-Controller hat ein Telegramm empfangen und im Empfangspuffer abgelegt. Das Telegramm muss vom Anwenderprogramm der SPS quittiert werden. Erfolgt dies nicht (SPS wird angehalten), wird das Telegramm über ein TimeOut gelöscht.
1	1	Das Anwenderprogramm der SPS hat das Telegramm verarbeitet und zeigt dies über das Setzen von EQ dem PDnetIP-Controller an.

Ablauf Empfangsvorgang

Schritt	EB	EQ	Zustand
1	0	0	Empfangspuffer ist frei
2	1	0	Es wurde ein Telegramm empfangen
3	1	1	Das Anwenderprogramm der SPS hat das Telegramm verarbeitet
4	0	0	Empfangspuffer ist wieder frei

Fehlerliste Ausgangsworte (_AWS und _AWE)

Wert	Bedeutung
0	Kein Fehler
104	Verbindung wurde geschlossen
105	Aktiver Verbindungsaufbau gescheitert
106	Keine Verbindung vorhanden – Sendedaten können nicht zugestellt werden, da der PDnetIP-Controller selbst keine Verbindung aufbauen darf (Kann in der Konfiguration über „Aktiver Verbindungsaufbau“ geändert werden)
107	Sendevorgang gescheitert
109	Die Zahl gleichzeitig ausführbarer Sendevorgänge von 12 PDU´s pro Verbindung wurde überschritten.
111	TimeOut Sendedaten
112	TimeOut Empfangsdaten
113	Empfangsdaten konnten nicht verarbeitet werden, da die Struktur der Empfangsdaten zuvor nicht registriert wurde. Strukturen müssen über den Baustein PDSTR (SFB692) beim PDnetIP-Controller registriert werden.
114	Sendedaten konnten nicht verarbeitet werden, da die Struktur der Sendedaten zuvor nicht registriert wurde. Strukturen werden mit dem Baustein PDSTR (SFB692) beim PDnetIP-Controller registriert.

Verbindungsstatus (_STATE)

Wert	Bedeutung
1	Verbindung wurde von der Partner-Station geöffnet. Dieser Status lieg vor, wenn die Verbindung im PDnetIP-Controller als passiv projektiert wurde.
2	Der PDnetIP-Controller baut eine Verbindung auf. Der Status steht solange an, bis der eigene CR mit einem CC der Partnerstation bestätigt wurde.
3	Der PDnetIP-Controller besitzt eine aktive Verbindung zur Partnerstation. Diese Verbindung wurde aktiv aufgebaut.
4	Es besteht keine Verbindung zu einer Partnerstation.

Die maximale Größe der TPDU´s beträgt 2.048 Byte. Baut der PDnetIP-Controller aktiv eine Verbindung auf, so wird im „Connection Request“ eine TPDU-Größe von 2.048 Byte festgelegt. Antwortet die Partnerstation im „Connection Confirm“ mit einer kleineren TPDU-Größe, nutzt der PDnetIP-Controller diese.

Baut eine andere Station aktiv eine Verbindung zum PDnetIP-Controller auf, nutzt dieser die im „Connection Request“ übermittelte TPDU-Größe, sofern diese nicht größer als 2.048 Byte ist. Andernfalls quittiert der PDnetIP-Controller im „Connection Confirm“ die TPDU-Größe mit 2.048 Byte und erwartet dass die Partnerstation diese Größe in den Sendedaten benutzt.

2.2.2.3 TCP-Schnittstelle

Der PDnetIP-Controller unterstützt **optional** eine freie TCP-Schnittstelle. Dazu benötigt er mit Hilfe des FB62 Zugriff auf den Signalspeicher der Steuerung.

Auf Netzebene werden dazu TCP-Daten direkt in den Signalspeicher der Steuerung übertragen.

Die zur Verarbeitung notwendigen Informationen werden vorab mit NetPro projektiert, und im nichtflüchtigen Setupdatenspeicher des PDnetIP-Controllers abgelegt.

Die TCP-Projektierungsinformationen werden als eine Liste von Verbindungen erstellt. Jede Verbindung besitzt folgende Eigenschaften:

- Eine Kennung (1..65535) zur eindeutigen Identifikation der Verbindung.
- Einen Namen (24 Zeichen) zur symbolischen Identifikation der Verbindung.
- Die IP-Adresse der Partnerstation sowie die zugeordnete TCP-Portnummer.
- Die genutzte Auftragsart (SEND, RECEIVE oder SEND und RECEIVE). Eine Verbindung mit der Auftragsart SEND kann nur Daten senden und keine Daten empfangen. Entsprechend kann eine Verbindung mit der Auftragsart RECEIVE nur Daten empfangen und keine Daten senden. Die Betriebsart SEND/RECEIVE ermöglicht eine bidirektionale Kommunikation.
- Die optionale Übertragungsart „Header mit Kennung“ die vor den Nettodaten einen Header sendet, welcher es der Partnerstation ermöglicht die Daten zu unterscheiden.
- Die Adresslage des Datenpuffers im Signalspeicher der Steuerung. Die Adresslage wird dabei für Strukturen als Kennung (symbolisch) und andernfalls im Wortbereich der Steuerung definiert. Wird die Option „Header mit Kennung“ genutzt wird eine Liste der Datenpuffer definiert.

Jede projektierte Verbindung belegt im Signalspeicher der Steuerung vier Merker, sowie acht Worte. Über diese Daten kommuniziert das Anwenderprogramm der SPS mit der TCP-Schnittstelle der Firmware.

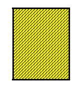
Zur Ablage dieser Informationen ist vorab eine Startadresse im Wort- sowie eine Startadresse im Merkerbereich der Steuerung zu definieren. Die Adressen werden von der TCP-Schnittstelle automatisch belegt.

Die folgende Tabelle beschreibt die Bedeutung der für jede Verbindung belegten Status-Worte und Merker. Der Suffix ergänzt den mit NetPro projektierten Verbindungsnamen.

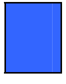
Suffix	Vorhanden	Zustand
‘_ID’	Immer	Die mit NetPro projektierte Verbindungskennung.
‘_STATE’		Der aktuelle Verbindungsstatus.
‘_SB’	Sende-richtung	Beauftragungsbit in Senderichtung. Über das Sendebit zeigt das Anwenderprogramm der SPS, dass ein Sendetelegramm vorliegt.
‘_SQ’		
‘_TXSTATE’		Letzter/Aktueller Status in Senderichtung.
‚_TXLEN’		Anzahl zu sendender Bytes
‚_TXSTRUC’		Kennung der zu sendenden Struktur *)
‘_EB’	Empfangs-richtung	Das Empfangsbit informiert das Anwenderprogramm der SPS über ein neues Empfangstelegramm.
‘_EQ’		
‘_RXSTATE’		Letzter/Aktueller Status in Empfangsrichtung.
‚_RXLEN’		Anzahl empfangener Bytes
‚_RXSTRUC’		Kennung der empfangenen Struktur *)

*) Diese Worte werden nur in der Übertragungsart „Header mit Kennung“ genutzt, in diesem Fall bezieht der PDnetIP-Controller die Längeninformationen weder aus RXLEN und TXLEN noch aus den projektierten Strukturen.

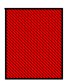
Die nachfolgende Legende zeigt die im folgenden genutzte farbliche Kodierung der Seite die für das Setzen des Zustandes verantwortlich ist.



Der Zustand wird vom PDnetIP-Controller gesetzt und dient dem Anwenderprogramm der SPS als Triggerbedingung.



Der Zustand wird vom Anwenderprogramm der SPS gesetzt und dient der Firmware des PDnetIP-Controllers als Triggerbedingung.



Der Zustand ist nicht zulässig und wird von einem fehlerhaften Anwenderprogramm der SPS verursacht.

Mögliche Zustände SB/SQ

SB	SQ	Zustand
0	1	Sendepuffer frei. Dieser Zustand wird vom PDnetIP-Controller auch im Einschaltmoment gesetzt, so dass im Anwenderprogramm der SPS keine Normierung der SB/SQ-Merker durchgeführt werden muss.
1	0	Wird vom Anwenderprogramm der SPS gesetzt, wenn ein Telegramm gesendet werden soll. Wurde der Sendevorgang angestoßen, darf dieser vom Anwenderprogramm der SPS nicht wieder gelöscht werden. In jedem Fall muss das Anwenderprogramm der SPS auf die Reaktion des PDnetIP-Controllers warten.
0	0	Zustand ist nicht zugelassen und wird vom PDnetIP-Controller auf SB=0 und SQ=1 normiert.
1	1	

Ablauf Sendevorgang

Schritt	SB	SQ	Zustand
1	0	1	Der Sendepuffer ist verfügbar. Der Zustand wird von dem PDnetIP-Controller initialisiert, auf diesen Zustand muss das Anwenderprogramm der SPS warten, um einen Sendevorgang starten zu können.
2	1	0	Das Anwenderprogramm der SPS hat den Sendepuffer mit Daten gefüllt und startet so den Sendevorgang.
3	0	1	Der PDnetIP-Controller hat den Sendewunsch bearbeitet und der Sendepuffer steht wieder zur Verfügung. Ob der Sendevorgang erfolgreich war, kann TXSTATE entnommen werden.

Mögliche Zustände EB/EQ

EB	EQ	Zustand
0	0	Empfangspuffer frei. Dieser Zustand wird vom PDnetIP-Controller im Einschaltmoment gesetzt, so dass im Anwenderprogramm der SPS keine Normierung der EB/EQ-Merker durchgeführt werden muss.
0	1	Zustand ist nicht zugelassen und wird vom PDnetIP-Controller normiert.
1	0	Der PDnetIP-Controller hat ein Telegramm empfangen und im Empfangspuffer abgelegt. Das Telegramm muss vom Anwenderprogramm der SPS, durch setzen des EQ-Bit, quittiert werden. Erfolgt dies nicht (SPS wird angehalten), wird das Telegramm über ein TimeOut gelöscht.
1	1	Das Anwenderprogramm der SPS hat das Telegramm verarbeitet und quittiert.

Ablauf Empfangsvorgang

Schritt	EB	EQ	Zustand
1	0	0	Empfangspuffer frei. Dieser Zustand wird vom PDnetIP-Controller im Einschaltmoment gesetzt, so dass im Anwenderprogramm der SPS keine Normierung der EB/EQ-Merker durchgeführt werden muss.
2	1	0	Der PDnetIP-Controller hat ein Telegramm empfangen und im Empfangspuffer abgelegt. Das Telegramm muss vom Anwenderprogramm der SPS durch setzen des EQ-Bit quittiert werden. Erfolgt dies nicht (SPS wird angehalten) wird das Telegramm über ein TimeOut gelöscht.
3	1	1	Das Anwenderprogramm der SPS hat das Telegramm verarbeitet und quittiert.
4	0	0	Der Empfangspuffer ist wieder verfügbar.

Sendestatus _TXSTATE

Wert	Bedeutung
0	Kein Fehler
20	Initialisierung beim Systemstart. Dieser Status wird einmal im Einschaltmoment gesetzt.
21	Sendedatenlänge ungültig Es wurde keine Länge oder eine zu große Länge in TXLEN geladen. Die maximale Länge wird in NetPro in der Projektierung des Sendepuffer festgelegt.
22	Der Sendevorgang ist gescheitert, obwohl im Output-Window der Verbindung für die Sendedaten genügend Platz vorhanden ist.
23	Der Verbindungsaufbau zur Partnerstation ist gescheitert.
24	Die Kombination von SB/SQ ist nicht zulässig. Werden SB/SQ zur Laufzeit in nicht zulässiger Art geändert, kann es zu nicht vorhersehbaren Kommunikationsproblemen führen. Die Ursache für diesen Fehler muss im Anwenderprogramm der SPS abgestellt werden.
25	SB/SQ wurde während eines laufenden Sendevorganges vom Anwenderprogramm der SPS geändert. Werden SB/SQ zur Laufzeit in nicht zulässiger Art geändert, kann es zu nicht vorhersehbaren Kommunikationsproblemen führen. Die Ursache für diesen Fehler muss im Anwenderprogramm der SPS abgestellt werden.
26	Der Sendepuffer kann nicht gelesen werden, da die Struktur zuvor nicht erfolgreich mit dem SFB692 beim PDnetIP-Controller registriert wurde.
27	Die Verbindung zur Partnerstation wurde geschlossen.
28	Beim Start des Sendevorganges durch das Anwenderprogramm der SPS (SB=1, SQ=0) initialisiert der PDnetIP-Controller einen Timer. Kann innerhalb dieser Zeit der Sendevorgang nicht erfolgreich abgeschlossen werden, bricht der PDnetIP-Controller den Sendevorgang ab. Der Timer ist von der Firmware-Version abhängig und beläuft sich in der Firmware 4.08.04.05 auf ca. 10 Sekunden.
29	Statusmeldung: Die Sendedaten liegen bereit und werden an die Partnerstation übertragen.
30	Statusmeldung: Der PDnetIP-Controller hat den Start eines Sendevorganges durch das Anwenderprogramm der SPS erkannt
31	Statusmeldung: Der PDnetIP-Controller wartet auf die Sendedatenlänge.
32	Statusmeldung: Der Sendevorgang wurde gestartet und der PDnetIP-Controller wartet auf den Sendepuffer.
33	Statusmeldung: Der Sendevorgang wurde gestartet und der PDnetIP-Controller wartet auf die Strukturkennung um den Sendepuffer zu bestimmen.
34	Die in Wort TXSTRUC übergebene Strukturkennung ist in der Projektierung nicht enthalten.
35	Die Kommunikation zur Partnerstation ist zu langsam. Der PDnetIP-Controller kann die Daten nicht so schnell zustellen, wie diese beauftragt werden.
36	Sendedatenlänge zu groß. Der PDnetIP-Controller kann in der Übertragungsart „Header mit Kennung“ maximal 510 Bytes senden.
37	Der Sendevorgang ist gescheitert, da keine Verbindung aufgebaut wurde und keine Partnerstation projektiert ist.
38	Optimiere Datenübertragung wurde projektiert aber der Datenpuffer ist noch ungültig.
39	Es ist kein Automatischer Verbindungsaufbau projektiert. Da keine Verbindung zu einer Partnerstation bestand wurde der Sendeauftrag negativ quittiert.

Empfangsstatus _RXSTATE

Wert	Bedeutung
0	Kein Fehler
50	Initialisierung beim Systemstart. Dieser Status wird einmal im Einschaltmoment gesetzt
51	Es liegen mehr Empfangsdaten vor als im Empfangspuffer abgelegt werden können. Es werden soviel Daten empfangen wie in den Empfangspuffer der SPS passen. Die verbliebenden Daten werden im nächsten Empfangszyklus im Empfangspuffer übergeben.
52	Die Verbindung zur Partnerstation wurde geschlossen.
53	Der Empfangspuffer kann nicht beschrieben werden, da die Struktur zuvor nicht erfolgreich mit dem SFB692 beim PDnetIP-Controller registriert wurde.
55	Die Kombination von EB/EQ ist nicht zulässig. Werden EB/EQ zur Laufzeit in nicht zulässiger Art geändert, kann es zu nicht vorhersehbaren Kommunikationsproblemen führen. Die Ursache für diesen Fehler muss im Anwenderprogramm der SPS abgestellt werden.
56	Beim Start des Empfangsvorganges durch den PDnetIP-Controller (Daten wurden empfangen) initialisiert der PDnetIP-Controller einen Timer. Kann innerhalb dieser Zeit der Empfangsvorgang nicht erfolgreich abgeschlossen werden, bricht der PDnetIP-Controller den Empfangsvorgang ab. Der Timer ist von der Firmware-Version abhängig und beläuft sich in der Firmware 4.08.04.05 auf ca. 10 Sekunden.
57	Statusmeldung: Der PDnetIP-Controller wartet auf das Durchschreiben der Empfangslänge RXLEN.
58	Statusmeldung: Der PDnetIP-Controller wartet auf das Durchschreiben der Empfangsdaten.
59	Die empfangene Strukturkennung ist in der Projektierung nicht enthalten. Daher ist es dem PDnetIP-Controller nicht möglich den Daten einen Empfangspuffer zuzuordnen. Die Verbindung wird daraufhin geschlossen.
60	Der Empfangsvorgang für den Header ist gescheitert, obwohl im Input-Window der Verbindung die Daten verfügbar sind.
61	Der Empfangsvorgang wurde abgebrochen, da ein leerer Sendepuffer (Größe=0) konfiguriert wurde.
62	Der Empfangsvorgang ist gescheitert, obwohl im Input-Window der Verbindung Daten vorhanden sind.

Die **markierten** Werte stehen nicht beim Abschluss eines Sende-/Empfangsvorganges an und dienen der Diagnose zur Laufzeit.

Verbindungsstatus (_STATE)

Wert	Bedeutung
1	Es besteht keine Verbindung zu einer Partnerstation.
2	Der PDnetIP-Controller baut eine Verbindung zur Partnerstation auf.
3	Der PDnetIP-Controller besitzt eine aktive Verbindung zur Partnerstation.
9	Verbindungsstatus unbekannt

Die Gesamtgröße der Auftragspuffer darf den zur Verfügung stehenden Arbeitsspeicher des PDnetIP-Controllers nicht überschreiten. Ein Auftragspuffer sollte die Größe von 4.096 Byte nicht überschreiten.

Ein Sendetelegramm wird nur dann übertragen, wenn bei der Partnerstation genügend Platz im Input-Window des Sockets zur Verfügung steht. Der PDnetIP-Controller arbeitet so, um eine Fragmentierung der Sendedaten zu vermeiden. Bedingt dadurch, dass bei einem „freien" TCP-Protokoll kein Anfang/Ende eines Telegramms von der Empfangsstation erkannt werden kann, nutzt der PDnetIP-Controller das beschriebene Verhalten. Kann das Telegramm nicht übertragen werden, generiert der PDnetIP-Controller den Fehlercode 35 und beendet den Sendevorgang.

Alle gesendeten/empfangenen Daten werden vom PDnetIP-Controller ohne Veränderung dem Sendepuffer entnommen bzw. im Empfangspuffer abgelegt.

Eine Manipulation der Byte-Order erfolgt nicht, alle Daten verbleiben in ihrem Format. Dies bedeutet, dass die Daten zwischen Socket und Signalspeicher 1:1 kopiert werden.

Wird eine Verbindung zu einer Port-Nummer aufgebaut zu der bereits eine Verbindung besteht, wird die vorhandene Verbindung geschlossen. Zu einem Zeitpunkt kann pro Portnummer nur eine aktive Verbindung vorhanden sein.

Übertragungsart „Header mit Kennung"

In der Übertragungsart „Header mit Kennung" erfolgt keine statische Zuordnung der Sende-/Empfangspuffer. Über die in der NetPro-Projektierung des TCP-Interfaces zu erstellenden Tabelle „Strukturen" entnimmt die Firmware alle benötigten Informationen wo die Sende-/Empfangsdaten in der SPS liegen.

Die Tabelle „Strukturen" beinhaltet für jeden Eintrag die folgenden Informationen:

- **Protokoll-Kennung**. Die Protokoll-Kennung ist die im Header übertragene Kennung (LB/HB), welche den Inhalt der Daten klassifiziert.
- **Struktur-Kennung**. Die Struktur-Kennung identifiziert die der Protokoll-Kennung zugeordneten Sende-/Empfangspuffer im

Signalspeicher der Steuerung. Die Strukturen müssen zuvor mit dem SFB692 (PDSTR) gegenüber dem PDnetIP-Controller angemeldet werden.

- **Größe**. Die Größe (in Byte) spezifiziert wie groß der über den Socket übertragene Datenbereich ist. Dadurch können mehrere hintereinander gesendete Telegramme zerlegt werden.

Der PDnetIP-Controller legt für jeden Eintrag einen Puffer in Sende- und Empfangsrichtung an. **Wichtig: Zu einem Zeitpunkt kann der gleiche Puffer in Empfangsrichtung nur einmal genutzt werden. Das Verhalten muss von den Partnerstationen gewährleistet werden.**

- **Empfangsvorgang**
  Empfängt der PDnetIP-Controller einen Header, schlägt er die im Header enthaltene Struktur-Kennung in der Tabelle „Strukturen" nach. Aus der Tabelle leitet sich die Adresslage des Empfangspuffers (Struktur-Kennung) ab. Das Anwenderprogramm der SPS kann dem Wort „RXSTRUC" entnehmen, welche Struktur empfangen wurde.
- **Sendevorgang**
  Beauftragt das Anwenderprogramm der SPS einen Sendevorgang, entnimmt der PDnetIP-Controller dem Wort „TXSTRUC" welche Struktur gesendet wird.

Die Längeninformation wird ebenfalls der Tabelle entnommen. Die Worte RXLEN und TXLEN haben in dieser Übertragungsart keine Bedeutung.

Achtung: In der Übertragungsart „Header mit Kennung" dürfen die gesendeten Strukturen maximal 510 Byte groß sein.

Option „Zeitoptimierte Kommunikation zum Endgerät“

In dieser Betriebsart werden pro SPS-Zyklus alle Sende- und Statusdaten aus dem Signalspeicher der Steuerung in den PDnetIP-Controller übertragen. Dies bedingt, dass alle Sende- und Statusdaten in einem Zyklus gelesen werden können und nur für ein begrenzt großes Datenmodell zur Verfügung stehen. Wie groß das Datenmodell sein darf, hängt von der Projektierung des Endgeräteinterfaces und einer evtl. vorhandenen ICOS-Projektierung ab.

Achtung: Wird die zulässige Größe überschritten arbeitet der Treiber fehlerhaft!

Option „Beauftragung ohne Quittungsbehandlung“

In dieser Betriebsart arbeitet der PDnetIP-Controller ohne Quittungsbit und es gelten andere Regeln für die Verwendung von SB/SQ sowie EB/EQ. Um diese Betriebsart nutzen zu können, muss die Option „Zeitoptimierte Kommunikation zum Endgerät“ aktiviert werden.

Am Anfang des SPS Zyklus müssen alle Sendebits gelöscht und am Ende des SPS Zyklus alle Empfangsbits gelöscht werden. Diese Betriebsart ermöglicht es dem PDnetIP-Controller pro SPS-Zyklus ein Telegramm zu senden. Das Anwenderprogramm der SPS bekommt keine Meldung darüber, ob ein Sendevorgang erfolgreich war.

Option „KEEP ALIVE senden“

Wird diese Option aktiviert, sendet der PDnetIP-Controller zyklisch ein KEEP-ALIVE an die Partnerstation und die Zeitüberwachung der Verbindung auf Inaktivität der Kommunikation wird deaktiviert. Eine in dieser Betriebsart geöffnete Verbindung kann somit solange geöffnet bleiben, wie die Partnerstation erreichbar ist. Andernfalls baut der PDnetIP-Controller nach einiger Zeit eine nicht genutzte Verbindung ab.

2.2.2.4 Kundenspezifische Treiber

Es sind in der Firmware des PDnetIP-Controllers weitere kundenspezifische Treiber implementiert. Die entsprechende Dokumentation erfolgt dann in den Kunden- bzw. projektspezifischen Dokumenten.

Benötigen Sie ein spezielles Protokoll oder eine Erweiterung einer vorhandenen Implementierung, dann sprechen Sie uns darauf an.

2.2.3 Serielle Protokolle

Der PDnetIP-Controller unterstützt eine Reihe serieller Protokolle zur Kommunikation mit anderen Geräten über offene Schnittstellen, sowie kundenspezifische Implementierungen als Sonderlösung.

2.2.3.1 Seab 1/F Master

Der PDnetIP-Controller unterstützt **optional** das „Seab 1/F Master“ Protokoll. Dazu benötigt er mit Hilfe des PDNIP-Bausteins Zugriff auf den Signalspeicher der Steuerung.

Das Protokoll „Seab 1/F Master“ ermöglicht die Kommunikation zu folgenden Stationstypen und dazu kompatiblen Stationstypen:

- Micro
- U120
- U250

Eine genaue Beschreibung erfolgt im „Seab 1/F Benutzerhandbuch für PDnetIP-Controller“. Im folgenden werden die Projektierungseinstellung sowie Leistungsparameter der Implementierung kurz dargestellt.

Mit der **lizenzpflichtigen** Vollversion von NetPro werden die Parameter des Treibers und das Datenmodell projektiert.

- **TV, Vorlaufzeit.** Zeitspanne, die nach der Aktivierung von RTS abläuft, bevor das erste Zeichen des Telegramms gesendet wird.
- **TN, Nachlaufzeit.** Zeitspanne, die nach dem Senden des letzten Zeichens abläuft, bevor RTS deaktiviert wird.
- **TP, Pausenzeit.** Zeitspanne nach Deaktivieren von RTS, in der die Stationen und deren Modems die Umschaltung zwischen Senden/Empfangen durchführen.
- **TWM, Wartezeit Master.** Antwortet die Unterstation nicht innerhalb von TWM, wird damit dem Master ein Fehler angezeigt. TWM muss größer als TP+TV sein, damit eine Unterstation ein gültiges Telegramm übertragen kann. Um eine zuverlässige Zustandserkennung zu ermöglichen, müssen TP, TV und TN bei Master und Unterstation identisch sein.
- **TVM, Verzögerungszeit Master.** Wurde vom Master ein fehlerhaftes Telegramm empfangen, so verzögert der Master den nächsten Sendevorgang um TVM, um der Unterstation den Fehler anzuzeigen.

- **Wiederholung im Fehlerfall.** Wird ein Pollaufruf nicht oder fehlerhaft beantwortet, so wird der Pollaufruf so oft wiederholt, bis eine gültige Antwort eintrifft. Mit der Wiederholungszahl wird dieser Vorgang auf eine maximale Anzahl Pollaufrufe beschränkt. Sinnvolle Werte liegen zwischen 0-3.

Für Unterstationen sind die A-Bytes 1 bis 126 und 127 gültig, wobei die virtuelle Unterstation 127 allein dem Senden von Broadcast an alle Unterstationen dient.

In die Poll-Liste können maximal 126 Unterstationen eingetragen werden, die vom Seab1F-Treiber in dieser Reihenfolge gepollt werden. Die Poll-Liste ist unsortiert, Unterstationen können mehrfach vorhanden sein.

Micro

Objekt	Anzahl / Datentyp
Meldungen	128 / Bit
Echtzeit Meldungen	keine
Echtzeit Meldungen aus dem Ringpuffer	keine
Echtzeit Meldungen aus dem Datenmodell	keine
Zählwerte	24 / Word
Umgespeicherte Zählwerte	24 / Word
Messwerte 8Bit	24 / Byte
Messwerte 16Bit	24 / Word
Verwaltungsmeldungen	48 / Bit
Stationsmeldungen	9 / Byte (APEX Datentyp)
Befehle	64 / Bit
Sollwerte	32 / Int. / Word (digital / analog)
Aufrufe	4 / Byte (APEX Datentyp)

U120

Objekt	Anzahl / Datentyp
Meldungen	256 / Bit
Echtzeit Meldungen	256 / Bit
Echtzeit Meldungen aus dem Ringpuffer	256 / Bit
Echtzeit Meldungen aus dem Datenmodell	256/ Bit
Zählwerte	63 / Word
Umgespeicherte Zählwerte	63 / Word
Messwerte 8Bit	64 / Byte
Messwerte 11/16Bit	63 / (11 Bit + Vz) Word
Verwaltungsmeldungen	48 / Bit
Stationsmeldungen	9 / Byte (APEX Datentyp)
Befehle	256 / Bit
Sollwerte	16/32 / Int./Word (digital/analog)
Aufrufe	4 / Byte (APEX Datentyp)

U250

Objekt	Anzahl / Datentyp
Meldungen	1024 / Bit (in Summe mit E. Meldungen)
Echtzeit Meldungen	1024 / Bit (in Summe mit Meldungen)
Echtzeit Meldungen aus dem Ringpuffer	1024 / Bit
Echtzeit Meldungen aus dem Datenmodell	1024 / Bit
Zählwerte	128 / Word
Umgespeicherte Zählwerte	128 / Word
Messwerte 8Bit	256 / Byte (in Summe mit 16Bit Mw.)
Messwerte 16Bit	256 / Byte (in Summe mit 8Bit Mw.)
Verwaltungsmeldungen	48 / Bit
Stationsmeldungen	9 / Byte (APEX Datentyp)
Befehle	512 / Bit Feld
Sollwerte	256 / Int. / Word (digital / analog)
Aufrufe	4 / Byte (APEX Datentyp)

Broadcast

Objekt	Anzahl / Datentyp
Aufrufe	4 / Byte (APEX Datentyp)

2.2.3.2 IS-Tester

Das IS-Tester-Protokoll basiert auf der Schnittstellenspezifikation in Version 037.1 vom 27.04.2005 und implementiert ein serielles Protokoll zur Kommunikation zwischen einem RPS und einem IS-Tester.

Der PDnetIP-Controller wickelt dabei die serielle Kommunikation ab und überträgt zwischen RPS und IS-Tester vollständige Telegramme in Sende- und Empfangsrichtung. Die Prüfsumme wird automatisch gebildet und geprüft. Das Anwenderprogramm der SPS kommuniziert über Sende- und Empfangsbit sowie Sende- und Empfangspuffer mit dem Treiber.

Der Treiber wurde nicht für einen Mischbetrieb geplant oder getestet!

Das IS-Tester-Protokoll darf zu einem Zeitpunkt nur auf einer seriellen Schnittstelle des PDnetIP-Controllers genutzt werden.

Werden zugleich weitere Treiber (incl. VDM) genutzt, muss gewährleistet sein, dass immer alle Daten in einem SPS-Zyklus gekoppelt werden können.

Die serielle Kommunikation erfolgt mit 38.400 Baud (8N1). Das Protokoll sieht kein Hardware-Handshake vor, was dazu führen kann, dass Datenverlust auftritt. In diesen Fällen kann zwangsläufig das geforderte Timing nicht eingehalten werden.

Die in der Spezifikation definierte Request/Response-Ablauf zwischen RPS und IS-Tester muss durch das Anwenderprogramm der SPS gewährleistet werden. Der PDnetIP-Controller prüft den Datenstrom nicht auf derartige Regelverletzungen. Jedes Setzen des Sendebit löst einen Sendevorgang aus.

Die Implementierung unterstützt die Quellkennungen 0x40 für den RPS sowie 0x80 für den IS-Tester. Ferner werden die Telegrammtypen 0xAA, 0x01 sowie 0xFA unterstützt.

In einem Testaufbau konnte bei einer SPS-Zykluszeit von 16ms pro SPS-Zyklus erfolgreich ein Telegramm gesendet werden.

2.2.3.3 Kundenspezifische Treiber

Es sind in der Firmware des PDnetIP-Controllers weitere kundenspezifische Treiber implementiert. Die entsprechende Dokumentation erfolgt dann in den Kunden- bzw. projektspezifischen Dokumenten.

Benötigen Sie ein spezielles Protokoll oder eine Erweiterung einer vorhandenen Implementierung dann sprechen Sie uns darauf an.

3 Installation

3.1 Einbau des PAD-740e

Zum Einsatz der Baugruppe PAD-740e wird diese in das Zentral-Rack der Simatic S7 Steuerung eingesetzt. Die obere und untere Befestigungsschraube muss angezogen werden, um einen sicheren Kontakt des PAD-740e mit der Backplane zu gewährleisten.

Die Steckplatznummer hat keinen Einfluss auf das Anwenderprogramm.

3.2 Einbindung des PAD-740e in HW Konfig

Anschließend wird der PAD-740e in die Bestückungsliste der Steuerung eingetragen. Dazu wird in HW Konfig auf dem gewähltem Steckplatz die Baugruppe IM 463-2 aus "Hardware Katalog/Simatic 400/IM 400/S5 Adapter" eingetragen.

Die Objekteigenschaften des IM 463-2 werden nicht editiert.

Beim Übergang vom Stop in den Run Zustand wird von der Steuerung der Hardwareausbau mit der Bestückungsliste verglichen. Wurde der PAD-740e fehlerhaft eingetragen, so lässt sich das Anwenderprogramm der Steuerung (in Abhängigkeit der Einstellung in "HW Konfig/CPUxx/Objekteigenschaften.../Anlauf") nicht starten.

3.3 Einbindung der PDnet-Bausteine in Step7

Der FB62 stellt mit Hilfe der Funktionen FC211, FC213 und FC216 die Verbindung zwischen CPU und PDnetIP-Controller für alle Datenübertragungen her, die über den VDM (VirtuellerDaten Manager) abgewickelt werden. Dieser Baustein erhält alle zum Datentransfer notwendigen Informationen vom PDnetIP-Controller, so dass die Parametrierung des Bausteins in verschiedenen Anlagen gleich sein kann.

Dazu muss der FB62 zyklisch, mindestens aber aller 0,7 Sekunden im Anwenderprogramm aufgerufen werden.

Der FB62 kann von Anwender umbenannt werden, um Überschneidungen mit anderen Bausteinpaketen zu vermeiden. Dabei sind Nummern kleiner gleich 255 zu verwenden. Die Bausteine FC211, FC213 und FC216 können dagegen nicht umbenannt werden.

Mit Hilfe des Step7–Managers der Simatic wird das von der APEX automation technologies GmbH gelieferte Step7 Beispiel Projekt (zip-file) dearchiviert. Zur Installation darf das Zip-File **nicht** mit anderen Programmen dekomprimiert werden.

Aus dem Beispielprojekt werden die PDnet-Bausteine über den Step7-Manager in das jeweilige Anwenderprogramm eingebunden.

3.4 Parametrierung der PDnet-Bausteine

Der PAD-740e belegt 4 Kacheln in der S7-400. Der Zugriff des PAD-740e auf diese Speicherbereiche ist exklusiv. Um Überschneidungen mit anderen Baugruppen zu verhindern, ist die Basisadresse des Kachelbereichs zwischen 0 und 252 in durch vier teilbaren Werten wählbar.

Die Basisadresse des Kachelbereichs wird am Funktionsbaustein FB62 am Parameter "iKachel" angegeben. Mit diesem Wert wird ebenfalls der Endgerätetreiber des PAD-740e in der Netzwerk projektierung NetPro ("Bearbeiten/Stationen/Teilnehmer x/DPM") versorgt.

Die Anzahl der vom Endgerätetreiber tatsächlich genutzten Kacheln kann ebenfalls mit NetPro unter "Bearbeiten/Stationen/ Teilnehmer x/DPM" projektiert werden. Diese bestimmt das Verhältnis zwischen Programmlaufzeit und Datendurchsatz des FB62. Die Programmlaufzeit ist außerdem von der verwendeten CPU abhängig. Sie beträgt in der CPU412 bis zu 20 ms pro Kachel.

Parameter des FB 62

Parameter	Datentyp	Art	Bedeutung
IKachel	Integer	Eingang	Erste Kachel des PAD-740
iCP_ERROR	Integer	Ausgang	Fehlercode des Bausteins

Fehlercode des FB 62

Fehlercode	Bedeutung
0	kein Fehler aufgetreten
1	kein PAD-740 in der angegebenen Kachel gefunden oder Endgerätetreiber des PAD-740 nicht aktiviert
2 – 15	Fehler im Kachelzugriff

4 Konfiguration

4.1 Projektierung mit Setup Daten

Die Projektierung der PDnetIP-Controller erfolgt vollständig mit NetPro. Dieses Programm ist ein Produkt der APEX automation technologies GmbH. Es ist für den Einsatz mit den Windows-Versionen NT/2000/XP vorgesehen.

Jede Konfigurationsänderung kann mit NetPro über die serielle Schnittstelle in den PDnetIP-Controller geladen werden. Ist der PDnetIP-Controller mit der entsprechenden Option freigeschaltet, so steht dafür der Netzwerkkanal ebenfalls zur Verfügung.

Die Setupdaten und Firmware werden voneinander unabhängig im PDnetIP-Controller gespeichert. Wird eine neue Firmware geladen, bleiben die Setupdaten erhalten. Ein erneuter Programmiervorgang ist nicht notwendig.

4.2 Verwendung des NetPro DB-Exports

Bei der Verwendung der Dienste RFC1006, ModBus-Master und TCP erwartet die Firmware des PDnetIP Controllers eine Struktur von Steuerungsinformationen im Anwenderprogramm. Für den PAD-740e können diese in Form eines Datenbausteins von NetPro erzeugt und exportiert werden.

Diese Datenbausteinbeschreibungsdateien werden in Step7 importiert und übersetzt, der Name des Datenbausteins wird in die Symboltabelle eingetragen.

Die folgende Beschreibung geht von der Verwendung von Step7, Version 5.3 und SP3 aus.

- Öffnen Sie den Quellen-Ordner Ihres Projektes. Wählen Sie unter „Einfügen/Externe Quelle“ die mit NetPro erzeugte *awl-Datei aus.
- Markieren Sie diese Datei im Quellen-Ordner. Übersetzen Sie den Baustein mit „Bearbeiten/Übersetzen“. Dabei wird ein Datenbaustein mit der entsprechenden Nummer im Bausteine-Ordner angelegt. Achtung, dabei wird nicht geprüft, ob ein Baustein mit dieser Nummer schon vorhanden war!
- Öffnen Sie den Symbol Editor und ordnen Sie dort dem Datenbaustein einen Namen zu.

4.3 Firmware

Die Firmware des PDnetIP-Controllers wird mit dem Programm DLSE32 in den PDnetIP-Controller geladen. Das Programm kann unter den Windows-Versionen NT/2000/XP eingesetzt werden.

Das Laden einer neuen Firmware erfolgt in der nachfolgend beschriebenen Reihenfolge.

- Lademodus des PDnetIP-Controllers aktivieren
- Serielle Verbindung zwischen PC und PDnetIP-Controller herstellen (YDL52).
- DLSE32 starten
- Firmware über den Menüpunkt "Firmware"/"Auswählen" selektieren.
- Download der Firmware über den Menüpunkt OnLine/Download.
- Nach den Download führt der PDnetIP-Controller einen Neustart durch.
- Anschließend sind die Setupdaten mit NetPro zu projektieren und in den PDnetIP-Controller zu laden. Dazu muss dieser erneut in den Ladezustand versetzt werden.

4.4 Lader

Der Lader befindet sich, wie die Firmware und die Setupdaten, auf dem PDnetIP-Controller im Flash-Memory. Jeder PDnetIP-Controller wird mit einem Lader ausgeliefert und ist ohne Lader nicht funktionsfähig.

In den Firmware-Versionen 1.xx sowie 2.xx sind die Setupdaten direkt in das Firmware-Image integriert. Ab Firmware-Version 3.xx ist die Firmware von dem Setup getrennt. Um eine Firmware ab Version 3.xx einsetzen zu können, wird dazu (minimal) die Lader-Version 2 benötigt. Die aktuelle Lader-Version 5 wird empfohlen, da diese auch große Firmware-Versionen laden kann.

Welche Lader-Version eingesetzt wird, können Sie mit DLSE32 bestimmen. Starten Sie dazu den PDnetIP-Controller im Lader und führen Sie die Funktion Online/Info aus.

Sollten Sie über einen älteren Lader verfügen und eine neue Version benötigen, so muss der PDnetIP-Controller zur Aktualisierung des Laders eingesandt werden.

Alternativ kann der Lader mit einer Spezial-Firmware aktualisiert werden. Der Einsatz dieser Firmware erfolgt auf eigenes Risiko. Eine Anwendung ohne Ersatzkarte ist in jedem Fall zu vermeiden. Scheitert die Aktualisierung, ist der PDnetIP-Controller nicht mehr funktionsfähig! Zur Reaktivierung muss der PDnetIP-Controller eingesandt werden.

4.4.1 Lader-Versionen

Die folgende Tabelle gibt eine Übersicht über die Baudrate mit welcher der Lader über die serielle Schnittstelle kommuniziert. Die genutzte Baudrate muss bei der Konfiguration von DLSE32 bzw. NetPro beachtet werden.

<table>
<tr><th>Lader-Version</th><th>COM1</th><th>COM2</th></tr>
<tr><td>1</td><td rowspan="5">19.200 Baud</td><td>19.200 Baud</td></tr>
<tr><td>2</td><td rowspan="4">115.200 Baud</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td><td>115.200 Baud</td><td>115.200 Baud</td></tr>
</table>

4.4.2 Lader-Aktualisierung

Die folgende Anleitung beschreibt die einzelnen Punkte zur Lader-Aktualisierung.

Beachten Sie die zuvor gemachten Hinweise zum möglichen Risiko einer Lader-Aktualisierung!

Ablauf der Lader-Aktualisierung:

1. Bringen Sie den PDnetIP-Controller mit Hilfe des roten Tasters in den Lader.
2. Laden Sie mit DLSE32 die Spezial-Firmware mit dem Dateinamen „P250LUP3.BIN“ in den PDnetIP-Controller.
3. Anschließend führt der PDnetIP-Controller einen Neustart durch und signalisiert über die Leuchtdioden 5..12 (gelber Block) den Programmiervorgang.
4. Der Abschluss der Programmierung wird durch das Leuchten der Leuchtdiode 8 angezeigt. Drücken Sie nun zum aktivieren des neuen Laders den Taster, bis die Leuchtdioden 5..12 (gelber Block) alle der Reihe nach eingeschaltet wurden.
5. Laden Sie jetzt eine Firmware in den PDnetIP-Controller. Mit dem Firmware-Download ist die Lader-Aktualisierung abgeschlossen.

5 Anwendungen

5.1 MODBUS/TCP

Um das Protokoll MODBUS/TCP nutzen zu können, muss der SF62 zyklisch im Anwenderprogramm der SPS aufgerufen werden. Der Baustein ist für die Kommunikation zwischen dem Signalspeicher der CPU und dem PAD verantwortlich. Unabhängig von der Art der Anwendung besitzt der Baustein keine treiberspezifischen Parameter.

Soll auf den Signalspeicher der S7-400 von außen zugegriffen werden, so wird die Server-Funktionalität benötigt. Dazu genügt es, den FB62 zyklisch aufzurufen. Weitere Anpassungen im Programm der SPS sind nicht notwendig.

In der „ModBus-Welt“ wird zwischen den Datentypen Register und Coil unterschieden. Ein Register ist ein Wort (16 Bit) und ein Coil ein Merker. Die Beschreibung verwendet für lokale Adressangaben im allgemeinen die Namen der SPS-Datentypen. Die Verwendung von Register und Wort sowie von Coil und Merker sind an einigen Stellen entsprechend austauschbar.

Soll die S7-400 selbst aktiv mit anderen MODBUS/TCP-Stationen im Netzwerk kommunizieren (Master-Funktionalität), sind folgende Punkte zu berücksichtigen:

- Die Adressen der ModBus-Auftragsblöcke sind in NetPro zu definieren.
- Für jeden ModBus-Auftragsblock ist ein Speicherbereich von 16 Worten in der SPS zu reservieren. Ein ModBus-Auftragsblock darf zwischenzeitlich nicht für andere Aufgaben (Zwischenspeicher) genutzt werden.
- Im Programm der SPS werden die Aufträge über die ModBus-Auftragsblöcke gesteuert.
- Die ModBus-Auftragsblöcke werden nach dem Aufruf des FB62 vom PAD ausgewertet und die entsprechenden Aktivitäten ausgelöst. Die Ergebnisse stehen je nach Bearbeitungszeit der Gegenstation entsprechend zeit verzögert im Signalspeicher der SPS zur Verfügung.

5.1.1 ModBus Auftragsblock

Die nachfolgende Tabelle enthält eine Übersicht über die Elemente eines ModBus-Auftragsblockes.

Wort	Kürzel	Bedeutung
1	Statuswort	Aus dem Statuswort geht der aktuelle Zustand des Auftragsblockes hervor. **1 = IDLE** Der Auftragspuffer ist frei. Dieser Status ist für die Auftragsverwaltung der SPS reserviert und wird vom PDnetIP-Controller nicht benötigt. Der Status muss nicht berücksichtigt werden. **2 = START** Dieser Status übergibt die Kontrolle an den PDnetIP-Controller. Zu diesem Zeitpunkt muss der gesamte Auftragsblock vollständig mit den richtigen Daten gefüllt sein. Das Anwenderprogramm der SPS darf die Werte solange nicht verändern, bis der PDnetIP-Controller die Verarbeitung des Auftragsblockes abgeschlossen hat. Dies wird durch den Status 4 signalisiert. **3 = RUN** Über diesen Status zeigt der PDnetIP-Controller, dass der Auftrag erkannt wurde. Während der gesamten Verarbeitung steht dieser Status an. **4 = DONE** Der PDnetIP-Controller hat den Auftrag abgeschlossen und das Ergebnis des Auftrages kann dem Fehlerwort (Wort 15) entnommen werden. Der Puffer kann jetzt wieder vom Anwenderprogramm der SPS erneut beauftragt werden.

Wort	Kürzel	Bedeutung
2	Funktionscode	**1 = Read Register** Der PDnetIP-Controller liest aus einer anderen ModBus/TCP Station Daten aus dem Register bereich und legt diese in einem Datenbaustein der S7 ab. **2 = Write Register** Der PDnetIP-Controller schreibt Daten aus einem Datenbaustein der S7 in den Register-Bereich einer anderen ModBus/TCP Station. **3 = Read Coils** Der PDnetIP-Controller liest aus einer anderen ModBus/TCP Station Daten aus dem Coil-Bereich und legt diese in einem Datenbaustein der S7 ab. **4 = Write Coils** Der PDnetIP-Controller schreibt Daten aus einem Datenbaustein der S7 in den Coil-Bereich einer anderen ModBus/TCP Station.
3	Datentyp	Reserviert
4	Erweiterte Adresse	Reserviert
5	Lokale Adresse	Register- bzw. Coilnummer, die über das lokale ModBus-Mapping eine Verbindung zu adressiertem Datenbaustein und Offset herstellt.
6	Anzahl	Anzahl zu bearbeitender Daten. Der Wert bestimmt somit die Anzahl zu lesender/schreibender Daten im Signalspeicher.
7	IP-Adresse 1	IP-Adresse des adressierten ModBus/TCP Servers
8	IP-Adresse 2	
9	IP-Adresse 3	
10	IP-Adresse 4	
11	UnitID	Die Unit-ID (ModBus/TCP Daten) wird im ModBus/TCP-Telegramm abgelegt und ist eventuell für die weitere Verarbeitung in der adressierten ModBus/TCP Station relevant. Dies betrifft jedoch nur Geräte mit Routing-Fähigkeit und muss der jeweiligen Gerätedokumentation entnommen werden.
12	TRID	Die Transaktions-ID (ModBus/TCP Daten) wird im ModBus/TCP Telegramm abgelegt und hat für die weitere Verarbeitung im PDnetIP-Controller keine Bedeutung. Über die Transaktion-ID kann ein ModBus/TCP Telegramm markiert werden, so dass es sich mit einem Netzwerk-Sniffer eindeutig identifizieren lässt. Am einfachsten kann dazu die Transaktions-ID vor jeder Beauftragung inkrementiert werden.
13	Entfernte Adresse	Registeradresse des adressierten ModBus/TCP Servers. Diese Adresse bestimmt den Speicher bereich (Register/Coils) auf den sich die Operation (Read/Write) bezieht.
14	Timeout	Reserviert
15	Fehlerwort	Nach Verarbeitung des Auftrages steht in diesem Wort der Abschlussstatus der Operation. Die Tabelle „Fehlercodes“ dokumentiert die hier möglichen Werte.
16	Info	Erweiterte Fehlerinformationen, welche für die spätere Verwendung reserviert sind.

Wird vom Anwenderprogramm der SPS ein Auftrag gestartet, dürfen keine Änderungen am Auftragsblock durchgeführt werden bis der PDnetIP-Controller im Statuswort den Wert 4 (DONE) ablegt hat. Ein Auftrag wurde erfolgreich durchgeführt, wenn im Fehlerwort 0 steht.

Fehlercode	Bedeutung
0	Kein Fehler
100	Ungültiger Funktionscode im Auftragspuffer übergeben.
101	Es konnte keine Verbindung zur ModBus/TCP Station aufgebaut werden oder die Verbindung wurde spontan geschlossen.
102	Der Funktionscode wird von der geladenen Firmware nicht unterstützt.
103	Bei der laufenden Kommunikation zur ModBus/TCP Station ist ein Fehler aufgetreten.

5.1.2 Beispiele

Nachfolgend werden einige Beispiele für die Anwendung der Master-Funktion gegeben. Die Beispiele beschreiben die Funktionalität und wie der Auftragsblock dazu belegt werden muss. Die im Anwenderprogramm der SPS notwendige Ablaufsteuerung im Statuswort wird nicht beschrieben. Für Testzwecke können die Auftragsblöcke auch über die OnLine-Liste gestartet werden.

5.1.2.1 Read Register

Im Beispiel werden aus der ModBus/TCP Station mit der IP-Adresse 192.168.1.33 aus dem Register 500, 10 Register gelesen und in der lokalen SPS auf Register 2000 abgelegt.

```
W1      Statuswort             2
W2      Funktionscode          1
W3      Datentyp               0
W4      Erweiterte Adresse     0
W5      Lokale Adresse         2000
W6      Anzahl                 10
W7      IP-Adresse 1           192
W8      IP-Adresse 2           168
W9      IP-Adresse 3           1
W10     IP-Adresse 4           33
W11     UnitID                 0
W12     TRID                   0
W13     Entfernte Adresse      500
W14     TimeOut                0
W15     Fehlerwort             0
W16     Info                   0
```

5.1.2.2 Write Register

Im Beispiel wird ein Schreibbefehl an den ModBus/TCP Server mit der IP-Adresse 192.168.1.33 ausgelöst. Es werden 25 Merkerworte von dem lokalen Register 100 an den ModBus/TCP Server ab Register 500 geschrieben.

```
W1      Statuswort              2
W2      Funktionscode           2
W3      Datentyp                0
W4      Erweiterte Adresse      0
W5      Lokale Adresse          100
W6      Anzahl                  25
W7      IP-Adresse 1            192
W8      IP-Adresse 2            168
W9      IP-Adresse 3            1
W10     IP-Adresse 4            33
W11     UnitID                  0
W12     TRID                    0
W13     Entfernte Adresse       500
W14     TimeOut                 0
W15     Fehlerwort              0
W16     Info                    0
```

5.1.2.3 Read Coil

Im Beispiel werden aus dem ModBus/TCP Server mit der IP-Adresse 192.168.1.33 ab dem Coil 800, 10 Coils gelesen und in der lokalen SPS ab Coil 55 abgelegt.

```
W1      Statuswort              2
W2      Funktionscode           3
W3      Datentyp                0
W4      Erweiterte Adresse      0
W5      Lokale Adresse          55
W6      Anzahl                  10
W7      IP-Adresse 1            192
W8      IP-Adresse 2            168
W9      IP-Adresse 3            1
W10     IP-Adresse 4            33
W11     UnitID                  0
W12     TRID                    0
W13     Entfernte Adresse       800
W14     TimeOut                 0
W15     Fehlerwort              0
W16     Info                    0
```

5.1.2.4 Write Coil

Im Beispiel wird ein Schreibbefehl an den ModBus/TCP Server mit der IP-Adresse 192.168.1.33 ausgelöst. Es werden 30 Merker ab Coil 500 an den ModBus/TCP-Server ab Coil 310 geschrieben.

```
W1      Statuswort            2
W2      Funktionscode         4
W3      Datentyp              0
W4      Erweiterte Adresse    0
W5      Lokale Adresse        500
W6      Anzahl                30
W7      IP-Adresse 1          192
W8      IP-Adresse 2          168
W9      IP-Adresse 3          1
W10     IP-Adresse 4          33
W11     UnitID                0
W12     TRID                  0
W13     Entfernte Adresse     310
W14     TimeOut               0
W15     Fehlerwort            0
W16     Info                  0
```

5.1.2.5 RFC1006 (ISO TP0 over TCP)

Im Beispiel wird die Ablaufsteuerung einer „Send Receive“ Verbindung dargestellt.

Zu beachten ist, dass das Senden auftragsbezogen oder in einem Zeitraster erfolgt. Im folgenden Beispiel wird dafür der Merker "Sendetakt" verwendet. Dieser kann in einem Weckalarm-OB gesetzt werden.

```
//------- Bausteinaufruf fuer den PAD-740e
      L     0
      T     #iKachel

       CALL  FB    62 , DB5
       iKachelBase:=#iKachel
       iCP_STATUS :=#iError

//------- Auswertung des Fehlerausgangs
      L     #iError
      L     W#16#0
      ==I
      SPB   ok                              //kein Fehler

      L     #iError
      T     "FB62Error"                     //Fehlercode sichern

      L     "FB62ErCounter"
      INC   1
      T     "FB62ErCounter"                 //Anz. Fehler zählen

      SPA   norx                   //Fehler -> keine Bearbeitung

//------- Bearbeiten der Sendeverbindung
ok:   U     "Sendetakt"
      U     "RFC1006StrgDB".TCP_SQ          //Sendung bearb.?
      UN    "RFC1006StrgDB".TCP_SB
      SPB   tx

      SPA   notx

tx:   R     "Sendetakt"

      L     DB11.DBW    0                   //Wert in Quelle
      INC   1
      T     DB11.DBW    0

      S     "RFC1006StrgDB".TCP_SB          //stoße Sender an
      R     "RFC1006StrgDB".TCP_SQ

//------- Bearbeiten der Empfangsverbindung---
notx: U     "RFC1006StrgDB".TCP_EB
      UN    "RFC1006StrgDB".TCP_EQ          //etwas empfangen?
      SPB   rx

      SPA   norx

rx:   L     "RxCouter"                 //zaehle Empfangsereignis
      INC   1
      T     "RxCouter"

      S     "RFC1006StrgDB".TCP_EQ          //quittiere Empfang

norx: NOP   0
```

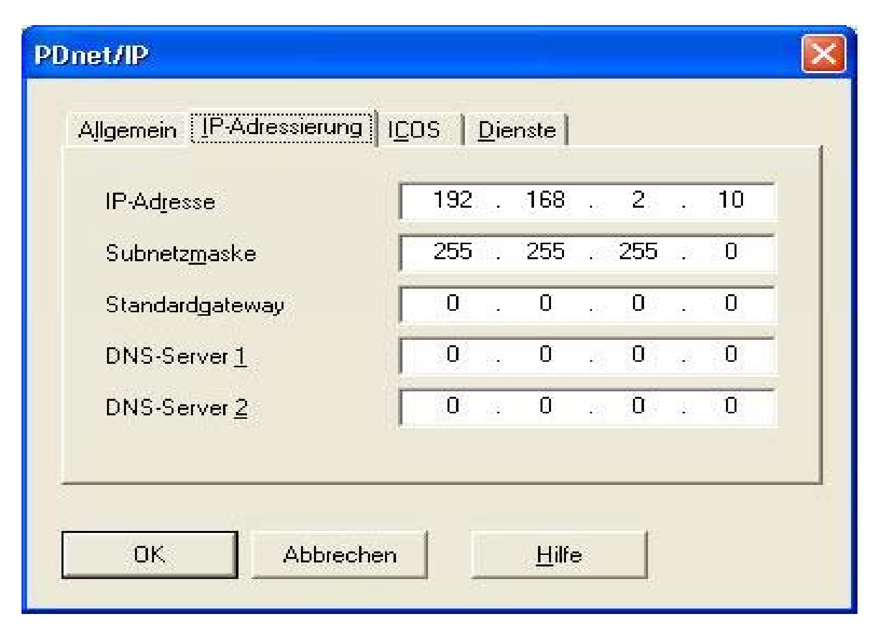

5.2 TCP-Schnittstelle

- Die Anwendung der TCP-Schnittstelle erfolgt analog zum RFC1006

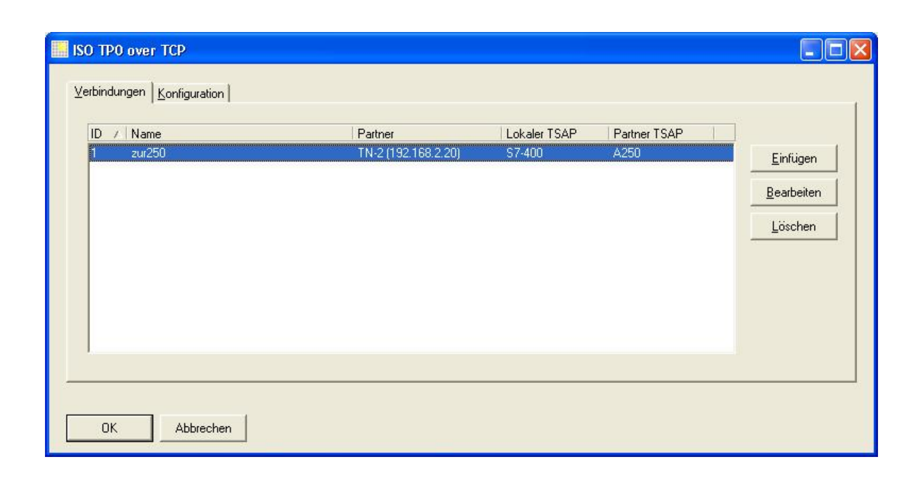
5.3 Projektierung mit NetPro

Das Kapitel zeigt Ihnen an einem Beispiel, wie eine neue RFC1006 Projektierung erstellt wird. Starten Sie dazu das Projektierungs-programm NetPro.

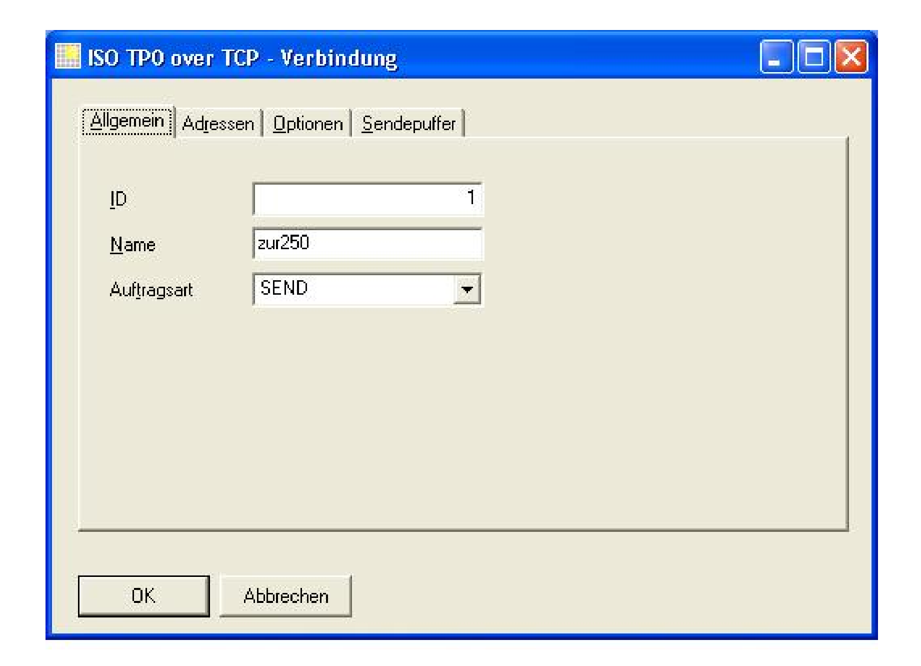
Erzeugen Sie ein neues Projekt und fügen Sie im Menüpunkt „Bearbeiten/Station" einen PAD-740e in das Projekt ein. Wenn Sie den eingefügten PDnetIP-Controller öffnen (Schalter „Bearbeiten"), erreichen Sie unter dem PDnetIP-Kanal den Bearbeiten-Dialog für den PDnetIP-Treiber, in welchem Sie die IP-Adresse und alle weiteren Netzwerk-Parameter definieren.



Über die Karteikarte „Dienste" erreichen Sie die Projektierung der RFC-Option, sowie aller weiteren Netzwerk-Dienste. Die Projektierung der RFC-Option erfolgt als Liste von Verbindungen.



Nachfolgend wird die Bedeutung der einzelnen Informationen beschrieben, welche für jede einzelne Verbindung projektiert werden.

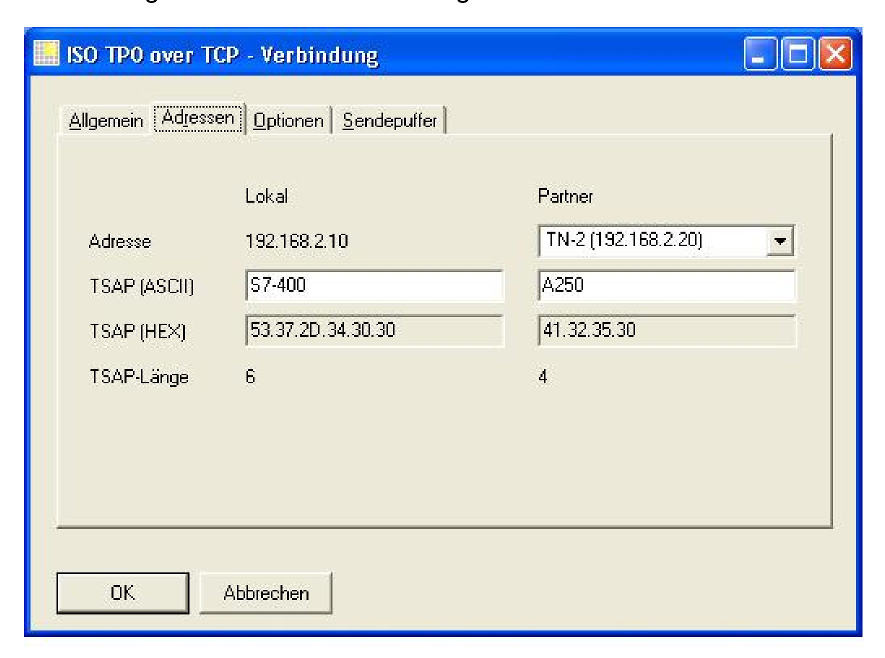


Die Kennung (1..65535), sowie der symbolische Name (24 Zeichen) dienen der eindeutigen Identifikation der Verbindung. Die Identifikation wird vom PDnetIP-Controller ausschließlich intern genutzt. Der Name stellt die Grundlage für die generierten Einträge in der ASD-Datei dar.

Die genutzte Auftragsart definiert die Richtung der Datenübertragung. Ein Verbindung mit der Auftragsart SEND kann nur Daten senden und keine Daten empfangen. Entsprechend kann eine Verbindung mit der Auftragsart RECEIVE nur Daten empfangen und keine Daten senden. Die Auftragsart SEND/RECEIVE ermöglicht eine bidirektionale Kommunikation.

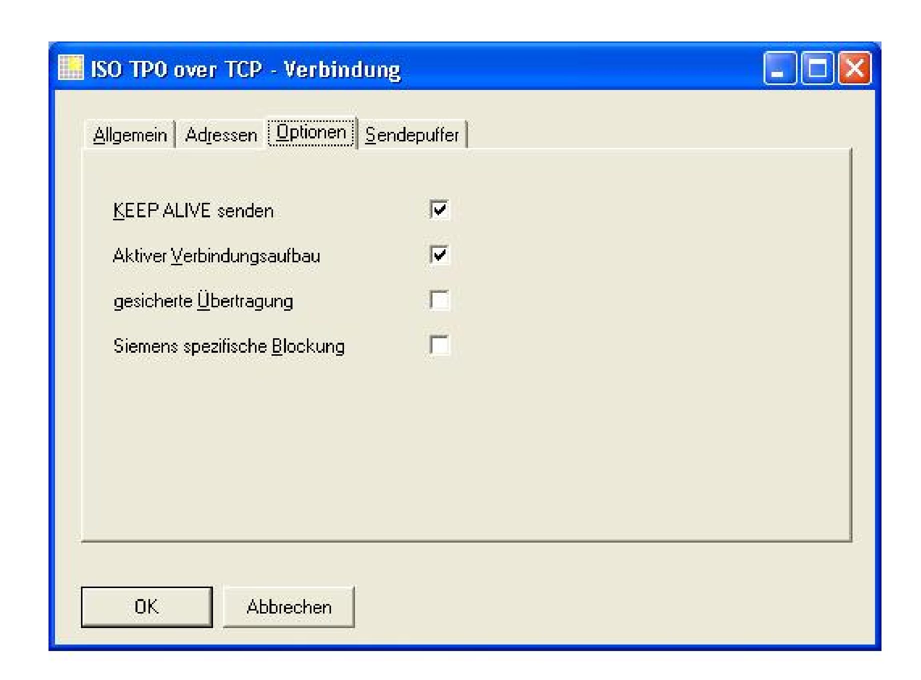
Wird in der Auftragsart RECEIVE keine IP-Adresse und kein TSAP definiert, wird diese Verbindung als unspezifizierte Verbindung

behandelt. Der PDnetIP-Controller akzeptiert dann einen Verbindungsaufbau von einer beliebigen Station.



Die IP-Adresse der Partnerstation identifiziert die Station, mit welcher der PDnetIP-Controller eine Verbindung aufbaut. Wird in den Optionen als Betriebsart RECEIVE gewählt, ist die Eingabe der Partnerstation optional. Der TSAP (Transport Service Access Point) dient der Identifikation der Verbindung auf beiden Seiten (Sender und Empfänger).

Um die Verbindung aufbauen zu können, muss der lokale (eigene) TSAP dem Partner-TSAP, welcher in der Partnerstation definiert ist, entsprechen. Die Länge eines TSAP's beträgt oft 8 Bytes, kann aber im Bereich von 2 Bytes bis 16 Bytes liegen. Der lokale und der Partner-TSAP können identisch sein, da die Verbindung durch die unterschiedlichen IP-Adressen eindeutig ist. Wenn zwei Stationen mehr als eine Verbindung unterhalten sollen, müssen sich auch die TSAP's unterscheiden.

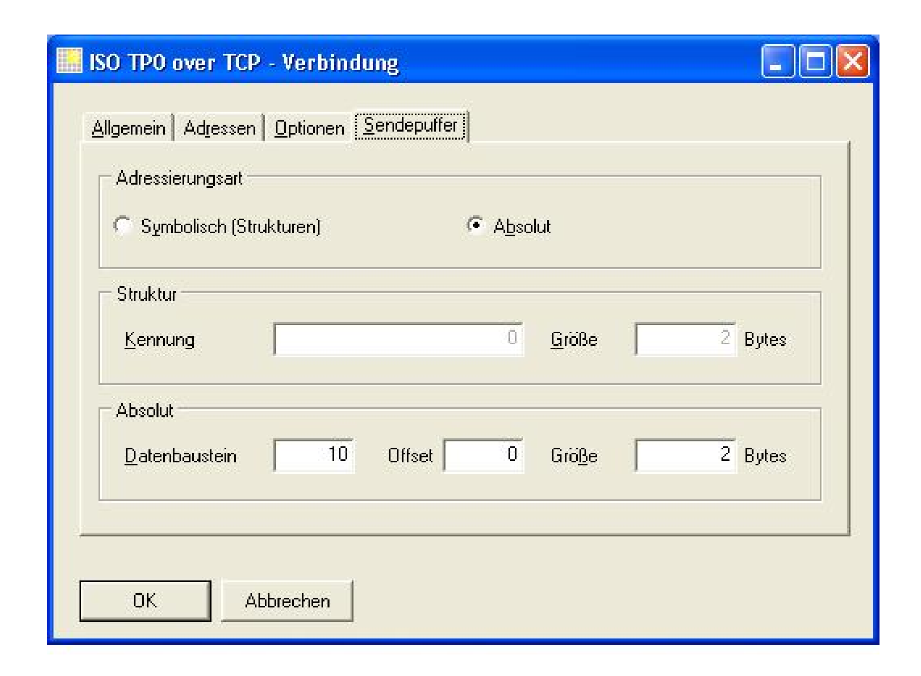


Wird die Einstellung „KEEP ALIVE senden“ aktiviert, sendet der PAD-250e zyklisch „KEEP ALIVE“ Telegramme. Ebenfalls wird die Verbindung nicht über ein Inaktivitäts-Timeout geschlossen, sondern nach einem nicht quittierten „KEEP ALIVE“.

Die Einstellung „Aktiver Verbindungsaufbau“ legt fest, ob die eigene Station aktiv den Verbindungsaufbau betreibt oder passiv den Verbindungsaufbau durch die Partnerstation erwarten soll. Auf beiden Seiten darf nicht die gleiche Einstellung definiert werden.

Der Übertragungsmodus „gesicherte Übertragung“ quittiert von der SPS gesendete Daten zu dem Zeitpunkt, wo alle Sendedaten das TCP Output-Window verlassen haben.

In der Übertragungsart „Siemens spezifische Blockung“ sendet der PAD-250e unabhängig von der, mit der Partnerstation ausgehandelten, TPDU-Größe nie mehr als 400 Byte pro TPDU.

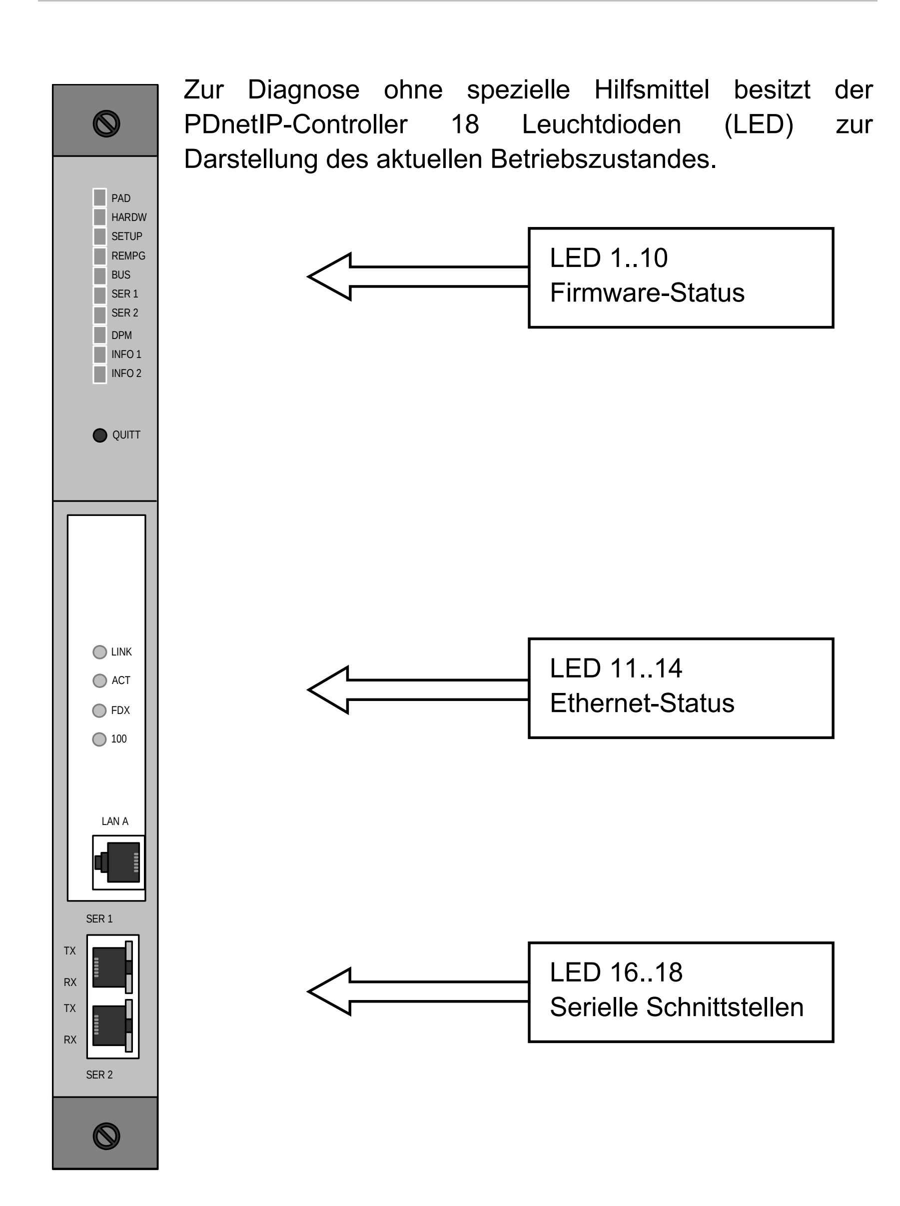


Die Adresslage der Sende-/Empfangspuffer definiert den Speicherbereich, in dem der PDnetIP-Controller die Empfangsdaten ablegt oder die Sendedaten entnimmt. Die Adresslage wird dabei für Strukturen als Name (symbolisch) und für andere Adressbereiche absolut als Signalspeicheradresse definiert.

6 Anzeige- und Bedienelemente

6.1 Status-Leuchtdioden

Zur Diagnose ohne spezielle Hilfsmittel besitzt der PDnetIP-Controller 18 Leuchtdioden (LED) zur Darstellung des aktuellen Betriebszustandes.



6 Anzeige- und Bedienelemente
6.1 Status-Leuchtdioden

6.1.1 Ethernet Status

Die vier grünen Leuchtdioden 11 bis 14 dienen der Diagnose des Hardware-Status der Ethernet-Schnittstelle. Die Steuerung der Leuchtdioden erfolgt direkt von der Hardware und ist somit von der Firmware unabhängig.

Ist eine LWL-Version des PDnetIP-Controllers weder über den MT-RJ- noch über RJ45- Anschluss angeschlossen, so toggelt die FDX LED im 5 Sekunden Rhythmus während der Suche nach einer Verbindung.

Nr.	LED	Zustand	Bedeutung
1	FDX grün	an	Voll-Duplex Verbindung
		aus	Halb-Duplex Verbindung
2	100 MBit grün	an	100-Mbit Verbindung
		aus	10-Mbit Verbindung
3	ACT grün	flackert	PDnetIP-Controller sendet oder empfängt Daten
		aus	Keine Datenübertragung
4	LINK grün	an	Verbindung vorhanden
		aus	Keine Verbindung vorhanden

6.1.2 Firmware Status

Die acht gelben Leuchtdioden 1 bis 10 dienen der Diagnose des Firmware-Status und werden ausschließlich durch die Firmware gesteuert. Die Bedeutung der einzelnen Leuchtdioden ist abhängig von der in den PDnetIP-Controller geladenen Projektierung. In weiteren Beschreibungen von Firmware-Modulen wird im allgemeinen folgende Tabelle herangezogen.

Nr.	LED	Zustand	Bedeutung
1	PAD-Sts. gelb	An	Normalzustand, kein Fehler
		Blinkt	Firmwaremodul ohne spezielle Diagnose LED meldet Fehler
2	Hardware gelb	Aus	Normalzustand, kein Fehler
		Blinkt	Firmware hat Hardware Fehler erkannt
3	Setup gelb	An	PDnetIP-Controller besitzt Setupdaten
		Aus	Keine Setupdaten vorhanden
		Blinkt	Setupdaten fehlerhaft
4	Remote PG gelb	Aus	Normaler Betriebszustand
		An	Es besteht eine Verbindung zwischen dem PDnetIP-Controller und einen Programmiergerät
		Blinkt	Es trat ein Fehler in dieser Verbindung auf
5	Bus Status gelb	An	LAN Kanal arbeitet fehlerfrei
		Blinkt	LAN Kanal gestört, keine Verbindung vorhanden
6	Ser.1 Status gelb	Aus	Es wurde kein Treiber für die serielle Schnittstelle geladen
		An	Es wurde ein Treiber für die serielle Schnittstelle geladen
		Blinkt	Fehler in der Kommunikation der seriellen Schnittstelle
7	Ser.2 Status gelb	Aus	Es wurde kein Treiber für die serielle Schnittstelle geladen.
		An	Es wurde ein Treiber für die serielle Schnittstelle geladen.
		Blinkt	Fehler in der Kommunikation der seriellen Schnittstelle
8	DPM Status gelb	An	Die Kommunikation mit der SPS arbeitet fehlerfrei.
		Aus	Es wurde kein Endgerätetreiber zur Kommunikation mit der SPS aktiviert.
		Blinkt	Die Verbindung zwischen PDnetIP-Controller und SPS war oder ist gestört.
9	Info1		Zustand toggelt nach jeder Kommunikation mit dem Endgerät
10	Info2		reserviert

6.1.3 Serielle Schnittstellen

Die Leuchtdioden 15 bis 18 dienen der Diagnose der seriellen Schnittstellen. Die Steuerung der Leuchtdioden erfolgt direkt von der Hardware und ist von der Firmware unabhängig. Der Datenempfang der seriellen Schnittstellen kann durch ein Aufleuchten der entsprechenden Empfangs-LED der seriellen Schnittstelle geprüft werden.

Nr.	LED	Funktion	Bedeutung
15	Ser.1 TD Rot	an	Es werden Daten gesendet
		aus	Es werden keine Daten gesendet
16	Ser.1 RD grün	an	Es werden Daten empfangen
		aus	Es werden keine Daten empfangen
17	Ser.2 TD Rot	an	Es werden Daten gesendet
		aus	Es werden keine Daten gesendet
18	Ser.2 RD grün	an	Es werden Daten empfangen
		aus	Es werden keine Daten empfangen

6.2 Diagnose-Taster

Der Taster hat verschiedene Funktionen, welche vom Zustand in dem sich der PDnetIP-Controller befindet, abhängig ist.

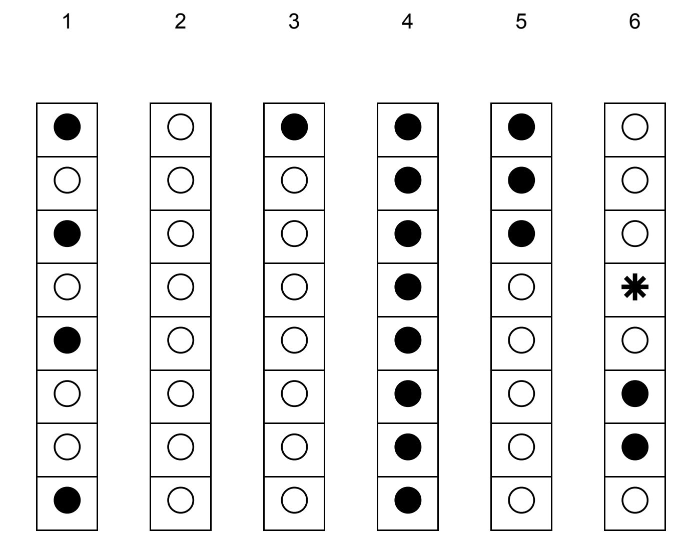
Funktion	Bedeutung
Fehler quittieren	Taster **kurz** drücken (ca. 1 Sek.) Die internen Fehlermerker der Firmware werden gelöscht. Blinkende LED's, deren Fehlerursache nicht mehr besteht, leuchten wieder dauernd.
Lader aktivieren	Die gelben LED's gehen der Reihe nach an bis alle acht LED´s an sind. Die Firmware führt anschließend einen Restart aus.
Lader verlassen	Der Koppler geht in den Lademodus über und es kann die Firmware über die seriellen Schnittstellen geladen werden. Remote-PG (blinkt), Seriell1 (ein) , Seriell2 (ein), alle anderen Gelben sind aus.

Wie kann erkannt werden in welchen Zustand sich der PDnetIP-Controller befindet?

Ist der Taster nicht gedrückt, kann dies am einfachsten an der Leuchtdiode Remote-PG erkannt werden. Ist diese LED aus, läuft eine Firmware.

6.2.1 Lader aktivieren

Die folgende Tabelle zeigt wie der Lader aktiviert wird wenn eine Firmware läuft. In jeder Spalte ist das Abbild der gelben Leuchtdioden 5..12 markiert.



Spalte	Bedeutung
1	Es läuft eine Firmware und es stehen keine Fehler an.
2	Der Taster wurde gedrückt und es gehen alle Leuchtdioden aus.
3	Im Sekundentakt gehen weitere Leuchtdioden an.
4	Nach acht Sekunden sind alle Leuchtdioden an, damit wurde die Firmware informiert, dass ein Neustart durchgeführt werden soll.
5	Der PDnetIP-Controller führt einen Neustart durch und es gehen wiederum alle Leuchtdioden der Reihe nach an
6	Anschließend befindet sich der PDnetIP-Controller im Lader was durch das blinken der Remote-PG Leuchtdiode erkennbar ist.

6.2.2 Lader verlassen

Die folgende Tabelle zeigt wie ein aktivierter Lader wieder verlassen werden kann.

1 2 3 4 5 6

Spalte	Bedeutung
1	Es läuft eine Lader.
2	Der Taster wurde gedrückt und es gehen alle Leuchtdioden aus.
3	Etwas schneller als im Sekundentakt, wandert die Leuchtdiode nach „unten“.
4	Alle gelben Leuchtdioden wurden durchlaufen..
5	Wenn die letzte gelbe Leuchtdiode erlischt führt der PDnetIP-Controller einen Neustart durch. In diesem Moment muss der Taster losgelassen werden, damit die Firmware starten kann.
6	Anschließend läuft im PDnetIP-Controller die Firmware.

7 Anhang

7.1 Fehlerliste TCP/IP-Stack

Die hier aufgeführten Fehlermeldungen sind bei der Diagnose hilfreich. Einige Firmware-Module (ICOS-Diagnose, Ausgaben der Diagnose-Firmware) protokollieren unter Umständen Fehlercodes des TCP/IP-Stack.

Fehler	Beschreibung
101	endpoint address not available
102	address in use
103	family not supported
104	arp table full
105	invalid baud rate
106	invalid comm port number
107	invalid device type
108	invalid interface number
109	invalid mask (ether must not be all fs)
110	invalid ping response
111	endpoint refused connection
112	destination address is required
113	destination unreachable (ICMP)
114	invalid parameter (pointer is 0)
115	interface closed
116	interface table full
117	interface open failed
118	operation (connect) is in progress
119	invalid function call (parameter)

7 Anhang
7.1 Fehlerliste TCP/IP-Stack

Fehler	Beschreibung
120	socket is already connected
121	multicast table full
122	multicast address not found
123	out of ports
124	network is down (send failed)
125	network unreachable (keepalive failed)
126	out of DCUs (packets)
127	option parameter is invalid
128	socket is not connected
129	RTIP not initialized
130	invalid socket descriptor
131	not enough devices
132	socket type or specified operation not
133	send failed due to output list
134	could not determine device
135	non-reentrancy error
136	routing table entry not found
137	routing table full
138	resource initialization failed
139	illegal operation due to socket shutdown
140	Timeout
141	type not supported
142	send needs to ARP but ARP is disabled
143	not enough heap memory for allocate request
144	table full (such as ARP cache, device table etc)
145	Illegal sized packet
146	device open failed
200	The global host table is full
201	The host table entry wasn't found
202	To many dns-servers
203	DNS server returned error
204	DNS name is too long
205	DNS call to socket function failed
206	data not available
207	socket call within DNS failed; probably No response received from server
208	No server in server table

